



Urheberrecht in der IT: Was Unternehmen und Freelancer oft falsch machen.



Warum 80 % IT-Freelancer und Unternehmen das Urheberrecht falsch verstehen - und dabei massiv Geld verlieren

Ein fiktiver Fall zeigt, wie schnell es teuer werden kann.

Die meisten IT-Freelancer glauben, dass sie mit der Berechnung ihres Stundensatzes an ein Unternehmen die Rechte an ihrem Code automatisch an den Auftraggeber abtreten.

Und die meisten Unternehmen glauben, dass sie nach Bezahlung der Honorarrechnung automatisch Eigentümer des gelieferten Codes sind.

Beides ist falsch!

Und zwar nicht ein bisschen, sondern fundamental.

Die rechtliche Realität in Deutschland ist:

Software ist ein urheberrechtlich geschütztes Werk.

Und der Urheber bleibt immer der Urheber (§§ 69a ff. UrhG).

Das gilt unabhängig davon, ob ein Vertrag existiert oder nicht.

Trotzdem kennen 80 % der Freelancer dieses Gesetz nicht – und 80 % der Unternehmen setzen sich täglich über diese Regeln hinweg, ohne es zu merken.

Warum ist das so?

Weil das Thema in der Branche kaum diskutiert wird und die meisten Dokumentationen juristisch, trocken und abstrakt sind.

Heute ändern wir das.

Dieser Artikel erklärt die Gesetzeslage verständlich – und zeigt anhand eines fiktiven, aber absolut realistischen Praxisfalls, wie schnell Unternehmen wie Freelancer in eine gefährliche Situation geraten.

1. Angestellte Entwickler vs. Freelancer

Angestellte Entwickler: Warum hier alle Rechte automatisch an den Arbeitgeber übergehen

Das deutsche Urheberrecht unterscheidet klar zwischen **angestellten Entwicklern** und **externen Entwicklern**.

Viele Menschen glauben: „Ein Entwickler ist ein Entwickler, egal ob angestellt oder extern.“

Falsch.

Juristisch besteht ein gigantischer Unterschied.

Angestellte Entwickler

Wenn ein Mitarbeiter im Rahmen seines Arbeitsvertrags Software schreibt, gilt grundsätzlich:

Alle Nutzungsrechte an dieser Software gehen automatisch auf den Arbeitgeber über.

Das ist üblich und notwendig, denn:

- die Firma trägt das wirtschaftliche Risiko,
- sie bezahlt Lohn & Arbeitsmittel,
- sie ist für Produkt, Haftung und Vermarktung zuständig.

Der Entwickler bleibt *zwar Urheber* (das kann niemand verlieren),
aber das Unternehmen erhält die **umfassenden Nutzungsrechte**, die es benötigt.

Freelancer und externe Entwickler

Bei Freelancern passiert dies NICHT automatisch.

Der Unterschied:

- Mitarbeiter: Rechteübertragung kraft Gesetz + Arbeitsvertrag
- Freelancer: Rechteübertragung nur durch ausdrückliche Vereinbarung

Das bedeutet:

Ein Unternehmen kann den Code eines Mitarbeiters verändern.

Den Code eines Freelancers aber nicht.

Und das ist der Punkt, an dem 80 % der Firmen unbewusst in die Falle laufen.

2. Das große Missverständnis: „Wir haben doch bezahlt – also gehört es uns.“

Dieser Satz ist in der IT so verbreitet wie falsch.

Bezahlung bedeutet **Nutzungsrecht**, nicht Eigentum.

Bezahlung bedeutet **interne Verwendung**, nicht Weiterentwicklung.

Bezahlung bedeutet **Zugriff**, aber niemals **Übernahme des Urheberrechts**.

Für Unternehmen hat das eine dramatische Konsequenz:

Wenn sie extern erstellten Code intern weiterbearbeiten, begehen sie juristisch eine Bearbeitung ohne Lizenz.

Das ist kein Kavaliersdelikt, sondern ein Verstoß gegen das Urheberrecht.

3. Das sagt das Gesetz - in einfachem Deutsch

Das deutsche Urheberrechtsgesetz ist überraschend klar:

1. Der Autor bleibt immer der Urheber

Er kann seine Urheberschaft nicht verlieren.

Auch nicht durch Bezahlung.

2. Der Kunde erhält nur die Rechte, die ausdrücklich eingeräumt wurden

Ohne Vertrag = nur einfaches Nutzungsrecht.

3. Der Kunde darf den Code nutzen - aber nicht verändern

Modifikation ist nur durch den Urheber oder durch diesen beauftragte Personen erlaubt.

4. Der Kunde darf den Code nicht weitergeben

Auch nicht intern an Tochterfirmen oder Partner.

5. Ohne Einräumung von Bearbeitungsrechten bleibt jede Änderung rechtswidrig.

4. Ein fiktiver Praxisfall (den fast jeder in der Branche kennt)

Ein externes Ingenieurbüro entwickelt für ein Unternehmen einen komplexen Java-Client zur Systemintegration.

Die Arbeit zieht sich über Monate – der Code ist sauber, dokumentiert und technisch anspruchsvoll.

Die Geschäftsführung weiß wenig darüber.

Das mittlere Management glaubt:

„Maier? Ach, das ist der externe Coder im Entwicklerteam. Kurz vor der Rente.“

Eines Tages verkündet der Manager der Abteilung:

„Wir haben ab nächstem Jahr weniger Budget.“

Wir setzen einen Junior auf den Code, der arbeitet das weiter.“

Was das Unternehmen nicht weiß:

- Der Code ist urheberrechtlich geschützt.
- Die Modifikation durch den Junior ist rechtswidrig.
- Jede Weiterentwicklung ohne Zustimmung verletzt die Lizenz.
- Der Hersteller haftet nicht für Schäden durch Fremdanpassungen.
- Das Unternehmen verliert im Zweifel sogar Supportansprüche.

Noch dramatischer:

Wäre der Junior im guten Glauben tätig, handelt trotzdem **das Unternehmen** fahrlässig.

Ein einziges professionelles Beratungsgespräch hätte dieses Risiko verhindert.

5. Die fünf größten Irrtümer in deutschen IT-Unternehmen

Irrtum 1: „Wir haben die Entwicklung bezahlt – also gehört uns der Quellcode.“

Falsch.

Sie haben ein Nutzungsrecht erworben, kein Eigentum.

Irrtum 2: „Unsere internen Entwickler dürfen den externen Code natürlich weiterbearbeiten.“

Falsch.

Das ist eine urheberrechtlich relevante Bearbeitung.

Irrtum 3: „Wir speichern die Tools einfach im GitHub-Repo, dann können alle weiterarbeiten.“

Das ist eine unzulässige Vervielfältigung und Weitergabe an Dritte.

Irrtum 4: „Der externe Entwickler hat nie etwas gesagt, also dürfen wir es.“

Nein.

Rechte entstehen automatisch durch das Gesetz.

Irrtum 5: „Das betrifft nur komplexe Software.“

Falsch.

Jede Klasse, jedes Script, jedes Bash-Tool ist urheberrechtlich geschützt.

6. Wie Freelancer jedes Jahr Tausende Euro verschenken

Viele Freelancer schreiben:

- keinen Lizenzheader
- keine Nutzungsrechte
- keine Regeln für Weiterentwicklungen
- keine AGB
- keine Einschränkung der Bearbeitung
- keine Hinweise auf Urheberrecht

Damit verschenken sie:

- Geld
- Einfluss
- zukünftige Beauftragungen
- Supportansprüche
- Wartungsverträge
- Eigentumsrechte
- Freelancer, die sauber mit Urheberrecht arbeiten, verdienen im Schnitt 20-40 % mehr – weil Unternehmen für Weiterentwicklungen zahlen müssen.

7. Was Unternehmen wissen müssen

Unternehmen handeln nicht böswillig.

In den meisten Fällen ist es Unkenntnis – oft sogar komplette Blindheit für das Thema.

ABER:

Unternehmen tragen das Haftungsrisiko.

Deshalb müssen sie:

1. verstehen, was ihnen gehört
2. verstehen, was sie dürfen
3. wissen, wo die Grenze liegt

4. wissen, wann der Urheber eingebunden werden muss
5. vertraglich klar regeln, welche Rechte sie benötigen
6. Gutes Engineering braucht gute Governance.

8. Fazit: Urheberrecht ist kein Streitpunkt - sondern der Schutzschirm der IT-Qualität

Das Urheberrecht ist nicht gegen Unternehmen gerichtet.
Es schützt die Integrität eines Systems.

Es sorgt dafür:

- dass Software konsistent bleibt
- dass Architektur nicht beschädigt wird
- dass Unternehmen planbar support erhalten
- dass kein Wildwuchs entsteht
- dass Qualität langfristig gesichert ist

Es ist das Sicherheitsnetz gegen Chaos.

Und genau deshalb sollten **Freelancer, Ingenieure, Unternehmer und Manager** dieses Thema unbedingt kennen.

9. Der Ausweg: Ein sauberer Lizenzvertrag schafft Klarheit für beide Seiten

Die gute Nachricht:

**Alle beschriebenen Probleme verschwinden sofort,
wenn ein sauberer Lizenz- und Nutzungsvertrag existiert.**

Ein Lizenzvertrag ist kein bürokratischer Aufwand,
sondern eine **technische Absicherung**,

die Missverständnisse vermeidet und Kosten spart.

Was gehört in einen guten Lizenzvertrag zwischen Freelancer und Unternehmen?

Ein professioneller Lizenzvertrag sollte mindestens folgende Punkte enthalten:

1. Urheberbenennung und Urheberrechte

„Der Entwickler bleibt gemäß §§ 69a ff. UrhG der Urheber des Werks.“

2. Art des Nutzungsrechts

- einfach oder ausschließlich
- zeitlich begrenzt oder unbegrenzt
- räumlich begrenzt (z. B. weltweit)
- inhaltlich begrenzt (z. B. interne Nutzung)

3. Regelung zur Weiterentwicklung (Kernpunkt!) „Eine Bearbeitung oder Weiterentwicklung des Quellcodes ist nur mit Zustimmung des Urhebers zulässig.“

Oder in der erweiterten Variante:

„Das Unternehmen erhält ein einfaches Bearbeitungsrecht unter der Voraussetzung, dass Änderungen dokumentiert und dem Urheber zur Kenntnis gebracht werden.“

4. Regelung zur Weitergabe an Dritte

- Tochterfirmen
- Partnerunternehmen
- externe Dienstleister
- Nearshore-Teams
- Cloud-Integratoren

Ohne ausdrückliche Erlaubnis verboten!

5. Support- und Wartungsregelung

Optional:

- Beauftragung des Urhebers für Weiterentwicklung
- Reaktionszeiten

- Stundensätze
- Übergabeprozesse

6. Updates und zukünftige Versionen

„Änderungen in Abhängigkeiten, Frameworks oder Security-Standards erfordern erneute Beauftragung des Urhebers.“

7. Haftungsklarheit

Der Urheber haftet nur für seine originäre Arbeit - nicht für spätere Fremdänderungen.

8. Dokumentation und Zugang zu Repositories

„Repos sind nur zur Nutzung bestimmt, nicht zur Weiterentwicklung ohne Lizenz.“

10. Sonderfall: Gemischte Teams - der Standardfall in modernen IT-Projekten

Das ist der schwierigste und wichtigste Teil.

Und fast niemand in der Branche weiß, wie man ihn rechtssicher löst.

Gemischte Teams bestehen aus:

- externen Freelancern
- internen Mitarbeitern
- externen Dienstleistern
- Nearshore-/Offshore-Teams
- Werkstudenten
- Praktikanten
- Subunternehmen

Der juristische Grundsatz lautet: **Wer Code schreibt, ist automatisch Urheber seines Teils.**

Das bedeutet:

1. Jedes Teammitglied ist Urheber seines Beitrags

Auch wenn es nur eine Klasse, ein Modul oder ein Script ist.

2. Der Auftraggeber benötigt daher Rechte von allen Urhebern

Wenn auch nur ein Urheber seine Rechte nicht ordnungsgemäß einräumt, kann der gesamte Code lizenzrechtlich „kontaminiert“ sein.

3. Der Auftraggeber darf nur anpassen, was er selbst hergestellt hat

Die externen Teile dürfen nur bearbeitet werden, wenn ein Bearbeitungsrecht vorliegt.

4. Unternehmen verlieren oft den Überblick

Viele Firmen haben keine Ahnung, wer welche Zeile Code geschrieben hat.

Das ist ein massives Compliance-Risiko.

11. Lösung: So regelt man gemischte Teams rechtssicher

Unternehmen benötigen drei Dinge:

A. Eine IP- und Lizenzrichtlinie für alle Teammitglieder

Ein Dokument, das festlegt:

- Wer Urheber ist
- Welche Rechte eingeräumt werden
- Was modifiziert werden darf
- Was nicht verändert werden darf
- Wer Reviews durchführen darf
- Wie Übergaben erfolgen

B. Einen Rahmenlizenzvertrag für externe Entwickler

Ein Vertrag, der:

1. Nutzungsrechte klärt
2. Bearbeitungsrechte klärt
3. Weitergabe verbietet oder erlaubt
4. Haftung und Support definiert
5. IP-Konflikte verhindert

C. Eine Repository-Governance

Hier werden festgelegt:

- Wer Commit-Rechte hat
- Wer Merge-Rechte hat
- Wer für fremden Code verantwortlich ist
- Wer nicht auf fremde Module zugreifen darf
- Wie Third-Party-Code gekennzeichnet wird

Beispiel für einen einfachen Governance-Satz:

„Interne Entwickler dürfen externe Module nicht bearbeiten. Änderungen erfolgen ausschließlich durch den Hersteller oder autorisierte Partner.“

12. Lösung für Freelancer: So schaffst du dir eine sichere Basis

Freelancer sollten IMMER folgendes tun:

1. Lizenzheader in jede Datei setzen
2. Nutzungsrechte explizit formulieren
3. Bearbeitungsrecht explizit ausschließen oder lizenzieren
4. AGB oder individuellen Lizenzvertrag verwenden
5. Dokumentieren, welche Teile von ihnen stammen
6. Sich im Projekt als technischer Verantwortlicher positionieren
7. Sicherstellen, dass Kunden bei gemischten Teams die Risiken verstehen
8. Für Weiterentwicklung separate Verträge anbieten

Wer das tut, schützt sich – und verdient langfristig mehr.

13. Lizenzmodelle - Die wirtschaftliche Grundlage jeder Software

Nicht jeder Code ist gleich wertvoll.

Nicht jede Anwendung dient dem gleichen Zweck.

Deshalb benötigen Softwarelizenzen immer ein zur **Anwendung passendes Modell**.

Hier sind die vier grundsätzlichen Anwendungsarten – und die dazu passenden Lizenzmodelle.

1. Anwendung nur für den Auftraggeber selbst (interne Nutzung)

Typisch für:

- interne Tools
- Test-Clients
- Validierungssoftware
- Automatisierungsskripte
- Monitoring
- interne Schnittstellen

Passende Lizenzmodelle:

A) Einmalzahlung + eingeschränktes Nutzungsrecht

Beispiel:

Der Kunde zahlt 10.000 Euro und nutzt die Software intern, aber darf sie nicht verändern.

B) Einmalzahlung + jährliche Wartungsgebühr

Für Updates, Framework-Anpassungen und Sicherheitskorrekturen.

C) Nutzungsrecht zeitlich begrenzt

1 Jahr, 3 Jahre, 5 Jahre

Industrie-Standard bei technischen Tools.

2. Anwendung wird Teil eines Produkts, das der Auftraggeber verkauft

Das ist wirtschaftlich eine völlig andere Liga.

Beispiel:

Ein Kunde lässt einen Algorithmus entwickeln, den er anschließend in seiner SaaS-Lösung einsetzt.

Hier muss der Urheber eine Lizenz einräumen, die dem Kunden erlaubt:

- die Software in ein Produkt einzubauen
- Geld damit zu verdienen

Passende Lizenzmodelle:

A) Hohe Einmalzahlung (Buy-out)

Teuer, weil der Urheber damit alle wirtschaftlichen Chancen abtritt.

B) Jährliche Vertriebs-/Produktionslizenz

Weit verbreitet in der Industrie.

C) Umsatzbeteiligung / Royalty

Der eleganteste und fairste Weg bei Produkten.

Beispiel:

5 % vom Umsatzanteil des funktionalen Moduls.

3. Kleine Tools, Hilfswerkzeuge, Nebenmodule

Diese Tools haben meist:

- geringen Funktionsumfang
- aber hohe technische Bedeutung

Zum Beispiel:

- Validierungstools
- Deployment-Skripte
- Data-Converter
- Konfigurationsgeneratoren

Passende Lizenzmodelle:

A) Kleine Einmalzahlung

B) Update-Vergütung nach Aufwand

C) Lizenz pro Projekt oder Team

3. Anwendung, die direkt Geld für den Auftraggeber erzeugt

Das ist der wichtigste und sensibelste Fall.

Beispiele:

- Matching-Algorithmus für Versicherung
- Pricing-Engine
- Fahrzeugbewertungen (wie manche DAT-Komponenten)
- Risk Engines
- Empfehlungslogiken

Wenn externe Software **Teil des Geschäftsmodells** wird,
dann kann der Urheber völlig andere Lizenzmodelle verlangen.

Passende Lizenzmodelle:

A) Umsatzbeteiligung

Standard in der Softwareindustrie.

Beispiel: 2 % der Erlöse, die durch das Modul entstehen.

B) Lizenz pro Nutzung / API-Call / Vorgang

Beispiel:

0,05 € pro Anfrage.

C) Lizenz pro Mandant

Typisch im B2B-Umfeld.

D) Lizenz pro Fahrzeug / Vorgang / Datensatz

Sehr verbreitet in Automotive-, Versicherungs- und Bewertungsportalen.

E) Mindestlizenz + variable Komponente

Beispiel:

20.000 € jährlich + 3 % vom Umsatzanteil.

Für viele Freelancer wäre das ein Vermögen –
sie wissen nur nicht, dass sie diesen Anspruch haben.

14. Erweiterter Schluss: Der Weg nach vorn

Urheberrecht ist kein Hindernis.

Es ist die Grundlage für:

- saubere Software
- klare Verantwortlichkeiten
- weniger Projektrisiken

- bessere Qualität
- weniger Konflikte
- sichere Budgets

Gute Software entsteht nur dort,
wo auch die rechtlichen Fundamentsteine sauber gesetzt sind.

Die Ingenieure von 4WT werden meist für Unternehmen aus dem DACH-Raum hinzugezogen wenn:

- Sie planen, den südostasiatischen Markt zu erschließen (Expansion, Markteintritt, Diversifizierung von Lieferketten);
- Sie deutsche Ingenieure mit technischem Sachverstand (juristische thailändische Person mit deutscher DNA), lokaler Präsenz und deutscher Management, jedoch keine Folienberatung suchen;
- Sie einen juristischen deutschen Partner vor Ort in Südostasien zur Unterstützung Ihrer Aktivitäten und als verlängerten Arm benötigen (operativ/technisch-exekutiv Betreuung vor Ort);
- Sie den Eindruck haben, dass in Ihrer Niederlassung ein Shadow-Management etabliert werden muss, welches die asiatische Arbeitskultur schon seit Jahrzehnten kennt, aber eine 100-prozentige deutsche DNA besitzt - Cultural Broker.
- Sie eine kulturell angepasste Unterstützung Ihrer thailändischen Geschäftsführung entsprechend deutscher Qualitäts- und Prozessanforderungen suchen.
- Sie die enormen Kosten ihres entsendeten Expat mit Sorge erfüllen und Sie eine deutlich günstigere skalierbare Alternative suchen.
- Sie eine Repräsentation bzw. bevollmächtigte Unternehmensvertretung als Liaison Office, Business Proxy oder Corporate Service Provider benötigen.
- Sie rechtssichere pragmatische FBA-konforme Inkubation suchen.

oder

- die Kosten für Ihre IT nicht mehr kalkulierbar sind und Sie keinen echten Mehrwert spüren,
- operativer Betrieb und IT nicht mehr sauber zusammenpassen und dadurch wirtschaftliche Entscheidungen ins Leere laufen,
- Sie das Gefühl haben, ihre Unternehmen arbeiten für die IT und nicht umgekehrt
- trotz gleicher Sprache es zu Verständnisschwierigkeiten zwischen Ihnen und Ihrer IT-Abteilung kommt und Sie einen Sparringspartner benötigen,
- Sie suchen eine Person, die ihnen in Augenhöhe ehrliche Antworten gibt und nicht aufgrund von Befürchtungen bezüglich seiner nächsten Beförderung oder seines

Arbeitsvertrags handelt.

- sich keiner Ihrer Mitarbeiter mehr mit den Legacy-(Alt-)Systemen im Keller auskennt,

Kein Vertrieb.

Kein Marketing.

reine Ingenieur-Analyse.

Austausch zwischen zwei Unternehmer, die Klartext reden.

Wenn Sie eine fachlich saubere Zweitmeinung brauchen:

☎ [+49 30 8687094010](tel:+49308687094010) (Bitte Zeitverschiebung nach Bangkok beachten.)

✉ uwe.richter@it-e-com.de

☞ [4WT Kontaktseite](#)

Ich sage Ihnen auch ehrlich, wenn kein Handlungsbedarf besteht.

Wem gehört der Code?

Urheberrecht in der IT – Ein teures Missverständnis für Unternehmen & Freelancer

80 % der IT-Freelancer & Unternehmen
80 %
...verstehen das Urheberrecht falsch. Sie glauben, mit der Bezahlung des Honorars gehen die Eigentumsrechte am Code automatisch auf den Auftraggeber über.

DAS IST FUNDAMENTAL FALSCH!

BEZAHLUNG ≠ EIGENTUM

Nach deutschem Recht (§§ 68a ff. UrhG) bleibt der Entwickler immer der Urheber seiner Software. Die Bezahlung gewährt dem Kunden standardmäßig nur ein einfaches Nutzungsrecht – nicht das Recht, den Code zu verändern, weiterzuentwickeln oder weiterzugeben.

RECHTLICHE REALITÄT: ANGESTELLTE VS. FREELANCER

ANGESTELLTE ENTWICKLER	FREELANCER & EXTERNE ENTWICKLER
Alle Nutzungsrechte an im Rahmen des Arbeitsvertrags erstellter Software gehen automatisch auf den Arbeitgeber über. Das Unternehmen darf den Code nutzen und verändern.	Es findet KEINE automatische Rechteübertragung statt. Ohne explizite vertragliche Regelung darf der Kunde den Code nur nutzen, aber NICHT verändern.

DIE 5 GRÖSSTEN IRRTÜMER & RISIKEN FÜR UNTERNEHMEN

IRRITUM 1: "Wir haben bezahlt, also gehört uns der Quellcode." Falsch. Sie haben nur ein Nutzungsrecht erworben, kein Eigen- oder Bearbeitungsrecht.	IRRITUM 2: "Unsere internen Entwickler dürfen den Code weiterbearbeiten." Falsch. Jede Änderung ohne explizite Erlaubnis ist eine Urheberrechtsverletzung.	IRRITUM 3: "Wir speichern alles in unserem Git-Repository, damit alle darauf zugreifen können." Falsch. Das ist eine unzulässige Weiterverbreitung und Weitergabe an Dritte.	IRRITUM 4: "Der Freelancer hat nichts gesagt, also ist es erlaubt." Falsch. Das Urheberrecht entsteht automatisch durch das Gesetz und muss nicht eingefordert werden.	IRRITUM 5: "Das betrifft nur grobe, kumulative Software." Falsch. Jedes einzelne Skript, jede Klasse und jedes Tool ist urheberrechtlich geschützt.
--	--	--	--	---

DAS VERLORENE POTENZIAL FÜR FREELANCER

Freelancer verschanken Geld, Einfluss und Aufträge. Ohne saubere Lizenzierung entgehen ihnen Einnahmen aus Wartungsverträgen, Support und bezahlten Weiterentwicklungen.

20-40 % MEHR VERDIENST MÖGLICH

Freelancer, die ihre Urheberrechte professionell managen und lizenzieren, verdienen im Schnitt deutlich mehr.

SOFORTMASSNAHMEN FÜR FREELANCER
Lizenzheader in jede Datei, AGB mit Nutzungsrechten verwenden und Bearbeitungsrechte explizit regeln (oder lizenzieren).

DIE LÖSUNG: DER SAUBERE LIZENZVERTRAG

Ein Lizenzvertrag ist eine technische Absicherung, kein bürokratischer Aufwand. Er schafft Klarheit für beide Seiten, vermeidet Missverständnisse und spart langfristig Kosten und Risiken.

WAS GEHÖRT HINEIN? (CHECKLISTE):

- ☑ **URHEBERBENENNUNG:** Klarstellen, wer der Urheber ist.
- ☑ **ART DES NUTZUNGSRECHTS:** Einfach/ausschließlich, zeitlich/räumlich begrenzt.
- ☑ **REGELUNG ZUR WEITERENTWICKLUNG:** Explizit erlauben oder ausschließen (der Kernpunkt!).
- ☑ **WEITERGABE AN DRITTE:** Regeln für Tochterfirmen, Partner etc.
- ☑ **HAFTUNG:** Der Urheber haftet nicht für Fremdänderungen.

Smarte Lizenzmodelle: Den Wert fair abrechnen

1. INTERNE NUTZUNG (Tools, Skripte) Modelle: Einmalzahlung mit eingeschränktem Nutzungsrecht, jährliche Wartungsgebühr; zeitlich begrenzte Lizenz.	2. CODE WIRD TEIL EINES VERKAUFTEN PRODUKTS Modelle: Hohe Einmalzahlung (Buy-out); jährliche Korbpreislizenz; Umsatzbeteiligung (Royalty).	3. CODE, DER DIREKT GELD GENERIERT (Pricing Engine, Algorithmen) Modelle: Umsatzbeteiligung; Lizenz pro Nutzung/API-Call; Lizenz pro Mandant/Datensatz.
---	--	--

URHEBERRECHT IST QUALITÄTSSICHERUNG.
Es schützt die Integrität der Software, sichert planbaren Support und verhindert unkontrollierten Wildwuchs.

Zufällige Auswahl von Artikeln zum Thema: Deutsches Recht

- [Warum Ingenieure oft die besseren Unternehmensberater für Geschäftsführer sind.](#)

In welchem Blogartikel kommt folgendes Suchwort vor?

Filter:

Suchen

Weiterführende Links

- [Freiberufler und freie Mitarbeiter richtig einsetzen – Scheinselbstständigkeit vermeiden](#)
- [Wenn IT-Projekte scheitern, ist selten die Technik das eigentliche Problem.](#)